ICESat (GLAS) Science Processing Software Document Series

Volume
GSAS Acceptance Test Procedures
Version 1.0

Jeffrey Lee/Raytheon ITSS
Observational Science Branch
Laboratory for Hydrospheric Processes
NASA/GSFC Wallops Flight Facility
Wallops Island, Virginia 23337

November 2000

ICESat Contacts:

H. Jay Zwally, ICESat Project Scientist
NASA Goddard Space Flight Center
Greenbelt, Maryland 20771

Bob E. Schutz, GLAS Science Team Leader
University of Texas Center for Space Research
Austin, Texas 78759-5321

David W. Hancock III, Science Software Development Leader
NASA/GSFC Wallops Flight Facility
Wallops Island, Virginia 23337

Foreword

This document contains the GLAS Science Algorithm Software (GSAS) Acceptance Test Procedures. This document is developed under the structure of the NASA STD-2100-91, a NASA standard defining a four-volume set of documents to cover an entire software life cycle. Under this standard a section of any volume may, if necessary, be rolled out to its own separate document. This document is a roll- out of the user guide within the Product Specification Volume.

The GEOSCIENCE LASER ALTIMETER SYSTEM (GLAS) is a part of the EOS program. This laser altimetry mission will be carried on the spacecraft designated EOS ICESat (Ice, Cloud and Land Elevation Satellite). The GLAS laser is a frequency-doubled, cavity-pumped, solid state Nd:YAG laser.

This document was prepared by the Observational Science Branch at NASA GSFC / WFF, Wallops Island, VA, in support of B. E. Schutz, GLAS Science Team Leader for the GLAS Investigation. This work was performed under the direction of David W. Hancock, III, who may be contacted at (757) 824-1238, hancock@osb.wff.nasa.gov (e-mail), or (757) 824-1036 (FAX).

The following GLAS Team members contributed to the creation of this document:

Raytheon / Anita Brenner

972 / David W. Hancock, III

Raytheon / Peggy Jester

Table of Contents

List of Figures

List of Tables

Section 1

Introduction

1.1 Identification of Document

This is the Acceptance Test Procedures document for the Version 1 delivery of the GLAS Science Algorithm Software (GSAS). The unique document identification number within the GLAS Ground Data System numbering scheme is *TBD*. Successive editions of this document will be uniquely identified by the cover and page date marks.

1.2 Scope of Document

The GLAS I-SIPS Data Processing System, shown in Figure 1-1, provides data processing and mission support for the Geoscience Laser Altimeter System (GLAS). I-SIPS is composed of two major software components - the GLAS Science Algorithm Software (GSAS) and the Scheduling and Data Management System (SDMS). GSAS processes raw satellite data and creates EOS Level 1A/B and 2 data products. SDMS provides for scheduling of processing and the ingest, staging, archiving and cataloging of associated data files. This document contains the Acceptance Test Procedures for the GSAS Version 1 delivery.

1.3 Purpose and Objectives of Document

The purpose of this document is to record the objectives, procedures, results and other technical information related to Acceptance Testing for the Version 1 delivery of GSAS.

1.4 Document Organization

This document's outline is assembled in a form similar to those presented in the NASA Software Engineering Program [Information Document 2.3a].

1.5 Document Status and Schedule

This is the initial version of the GSAS Acceptance Test Procedures for the V1 delivery.

1.6 Document Organization

This document's outline is assembled in a form similar to those presented in the NASA Software Engineering Program [Information Document 2.3a].

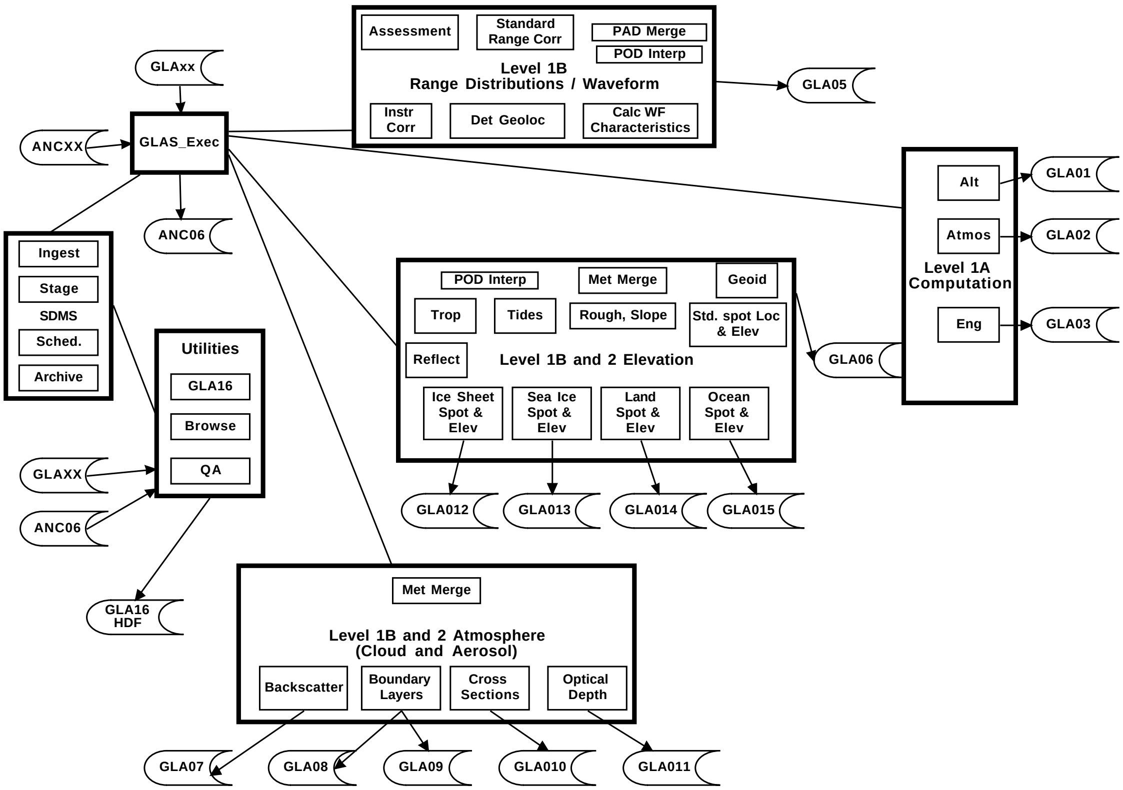

Figure 1-1 I-SIPS Software Top-Level Decomposition

1.7 Document Change History

Document Name: GLAS Science Algorithm Software Acceptance Test Procedures		
Version Number	**Date**	**Nature of Change**
Version 0	July 1999	Original Version.
Version 1	November 2000	Revised for V1 software.

Section 2

Related Documentation

2.1 Parent Documents

Parent documents are those external, higher-level documents that contribute information to the scope and content of this document. The following GLAS documents are parent to this document.

a) *GLAS Science Software Management Plan* (GLAS SSMP), Version 3.0, August 1998, NASA Goddard Space Flight Center, NASA / TM-1999-208641 / VER3 / VOL1.

The GLAS SSMP is the top-level Volume 1 (Management Plan Volume) document of the four volumes of NASA software engineering documentation [Applicable Reference 2.2c]. It dictates the creation and maintenance of the Product Specification Volume (Volume 2). This document is a roll out of the Product Specification Volume.

2.2 Applicable Documents

a) NASA Software Documentation Standard Software Engineering Program, NASA, July 29, 1991, NASA-STD-2100-91.

b) GLAS Science Algorithm Software Detailed Design Document, Version 1.0, October 20000, NASA Goddard Space Flight Center.

c) GLAS Science Algorithm Software User's Guide, Version 1.0, October 20000, NASA Goddard Space Flight Center.

d) GLAS Science Algorithm Software Version Description, Version 1.0, October 20000, NASA Goddard Space Flight Center.

e) GLAS ISIPS Operational Procedures Manual, TBD.

2.3 Information Documents

a) GLAS Level 0 Instrument Data Product Specification, Version 2.2, March 17, 1998, NASA Goddard Space Flight Center Wallops Flight Facility, GLAS-DPS-2610.

b) GLAS Standard Data Products Specification - Level 1, Version 2.0, January, 1999, NASA Goddard Space Flight Center Wallops Flight Facility, GLAS-DPS-2621.

c) GLAS Standard Data Products Specification - Level 2, Version 2.0, January, 1999, NASA Goddard Space Flight Center Wallops Flight Facility, GLAS-DPS-2641.

d) GLAS Science Data Management Plan (GLAS SDMP), Version 4.0, June 1999, NASA Goddard Space Flight Center Wallops Flight Facility, GLAS-DMP-1200.

Section 3

Test Identification and Objective

3.1 Objective

Acceptance testing of GSAS is intended to verify that the "ready-to-be-delivered software" satisfies all the requirements as stated in or derived from the software requirements documentation. Version 1 acceptance testing will cover a subset of the requirements documented in the GLAS Science Software Requirements Document (GSSRD).

Acceptance testing will repeat tests carried out during Integration testing, using the same input data files and like control files that define the execution scenarios. A set of reference output files, which have been created and verified during Integration testing, will be used to validate like files created during Acceptance testing.

3.2 Test Identification

There will be five main tests that execute scenarios matching requirements defined in the GSSRD. The results will be compared with the results of the corresponding integration test cases. The following scenarios will be executed as required for Version 1:

Table 3-1 Test Identification

Test	Description	Associated Requirements
1	Level 1A processing. Starting with GLA00, the software will create GLA01 and GLA02. Multiple granules of GLA01 and GLA02 will be created from a single GLA00.	GSDS-01400, GSDP-31000, GSDP-30200, GSDP-30201, GSDP-30202, GSDP-30203, GSDP-30210, GSDP-30600, GSDP-30601, GSDP-30602, GSDP-31200, GSDP-31800
2	Waveform processing. Starting with GLA01, the software will create GLA05. Multiple granules of GLA05 will be created from multiple granules of GLA01.	GSDS-01400, GSDP-31000, GSDP-30200, GSDP-30205, GSDP-30210, GSDP-30700, GSDP-30701, GSDP-30703, GSDP-31200, GSDP-31800
3	Atmosphere processing. The software will produce GLA07, 08, 09, 10, and 11 starting with GLA02 as input. A single granule (each) of GLA07-11 will be created from a single granule of GLA02.	GSDS-01400, GSDP-31000, GSDP-30200, GSDP-30207, GSDP-30208, GSDP-30210, GSDP-30700, GSDP-30704, GSDP-30705, GSDP-30800, GSDP-30806, GSDP-30807, GSDP-31200, GSDP-31800

Table 3-1 Test Identification (Continued)

Test	Description	Associated Requirements
4	Elevation processing. Starting with GLA05 product created in Test 2, the software will create GLA06, 12, 13, 14, and 15.	GSDS-01400, GSDP-31000, GSDP-30200, GSDP-30206, GSDP-30209, GSDP-30210, GSDP-30700, GSDP-30702, GSDP-30703, GSDP-30800, GSDP-30801, GSDP-30802, GSDP-30803, GSDP-30804, GSDP-30805, GSDP-31200, GSDP-31800
5	Partial Processing. The software will produce a GLA06 product starting with a GLA05 input produced from Test 2.	GSDS-01400, GSDP-31000, GSDP-30200, GSDP-30206, GSDP-30209, GSDP-30210, GSDP-30700, GSDP-30702, GSDP-30703, GSDP-30800, GSDP-30805, GSDP-31200, GSDP-31800

3.3 Deviations from Integration Testing

During the course of preparing for Acceptance testing, several GLA and ANC filenames were changed to more accurately reflect the GLAS file naming conventions. All files are clearly identified in the next section.

3.4 File Inventory

This section identifies the various files and directories used in GSAS Acceptance Testing. The runtime and source directories are fully described in the GSAS Version Description document.

The main acceptance test directory is located in $GLAS_HOME/test. The content of $GLAS_HOME/test is described in Table 3-2.

Table 3-2 GLAS_HOME/test

Item	Description
bin	Directory containing shell scripts unique to acceptance testing.
setup_acc_test.sh	Shell script which links necessary files into each acceptance test directory. Will also verify the existence of the needed files in the source directories. (This is actually a link to the script of the same name within the bin directory).
test1	Directory for testing the L1A scenario (Test 1).
test2	Directory for testing the Waveforms scenario (Test 2).
test3	Directory for testing the Atmosphere scenario (Test 3).

Table 3-2 GLAS_HOME/test (Continued)

Item	Description
test4	Directory for testing the Elevation scenario (Test4).
test5	Directory for testing the Elevation partial-processing scenario (Test 5).
test_data.	Directory where test GLA, dynamic ANC, and reference files are stored.

The bin directory contains scripts needed during the acceptance testing process. This is different from the GLAS_HOME/bin directory, which contains the actual GSAS binaries.The scripts within $GLAS_HOME/test/bin are described in Table 3-3.

Table 3-3 GLAS_HOME/test/bin

Item	Version	Description
setup_acc_test.sh	1.0	Shell script which links necessary files into each acceptance test directory. Will also verify the existence of the needed files in the source directories. It is linked into the $GLAS_HOME/test directory.
cleanup.*	n/a	This script will remove the outputs from the current run. Each script is linked into the respective runtime test directory.
verify.*	n/a	This script will verify the outputs of the test against reference binaries. If differences are found, the script will run the respective product_reader and create human-readable output files which may be “diffed” to determine the problem. Each script is linked into the respective runtime test directory.

The test1-test5 directories are “runtime” directories which contain links to the binary, library, and data files needed in each respective acceptance test. The tester will perform each acceptance test within the respective test directory. The test directories are created by setup_acc_test.sh and, if removed, can be completely re-created by running the script.

The test_data contains sample product data, sample dynamic ancillary data, requisite control files, and reference data. The contents are listed in Table 3-4. Detailed descrip-

Table 3-4 GLAS_HOME/test/test_data

Item	Description
00checksums	Results of the UNIX cksum command for all files in the directory.
anc01_001_20000101*.dat	Meteorological file data generated by a development-team reformatting utility. Original data from August 1999 but relabeled to Jan 1, 2000, in house. Originally from NCEP (ftp.ncep.noaa.gov directory: /pub/data/grib/fnl)
anc08_001_20000101_000000_01_00.dat	Precision Orbit Data file, in house, based on Bob Schutz's 97 min orbit. Starts on July 15, 2001, 75613 sec from midnight, duration=5820 sec.
anc09_001_20000101_000000_01_00.dat	Precision Attitude Data file, in house from Bob Schutz.
anc20_001_20000101_000000_01_00.dat	Predicted orbit file. Provided by science team.
anc24_001_20000101_120000_01_00.dat	Rotation matrix. Provided by science team.
cf01_001_20001027_001.ctl	Control file for Test 1.
cf02_001_20001027_001.ctl	Control file for Test 2.
cf03_001_20001030_001.ctl	Control file for Test 3.
cf04_001_20001027_001.ctl	Control file for Test 4.
cf05_001_20001027_001.ctl	Control file for Test 5.
gla00_001_20000101_*.dat	GLA00 data generated in-house by the development team. This data is not suitable to run outside of the L1A testing scenario.
gla01_001_11_001_0001_*_00.dat	GLA01 data created by the development team using scientifically-plausible values. Used as input to Test 2. See Section 3.4.1 for a detailed description.
gla01_reference_*.dat	GLA01 reference data created and verified during integration testing. The GLA01 data created in Test 1 should match this data.
gla02_001_11_001_0001_*_00.dat	GLA02 data created by the development team using scientifically-plausible values. Used as input to Test 3. See Section 3.4.2 for a detailed description.
gla02_reference_*.dat	GLA02 reference data created and verified during integration testing. The GLA02 data created in Test 1 should match this data.
gla03_reference_*.dat	GLA03 reference data created and verified during integration testing. The GLA03 data created in Test 1 should match this data.

Table 3-4 GLAS_HOME/test/test_data (Continued)

Item	Description
gla05_reference_*.dat	GLA05 reference data created and verified during integration testing. The GLA05 data created in Test 2 should match this data.
gla06_reference_*.dat	GLA06 reference data created and verified during integration testing. The GLA06 data created in Test 4 and Test 5 should match this data.
gla07_reference_*.dat	GLA07 reference data created and verified during integration testing. The GLA07 data created in Test3 should match this data.
gla08_reference_*.dat	GLA08 reference data created and verified during integration testing. The GLA08 data created in Test 3 should match this data.
gla09_reference_*.dat	GLA09 reference data created and verified during integration testing. The GLA09 data created in Test 3 should match this data.
gla10_reference_*.dat	GLA10 reference data created and verified during integration testing. The GLA10 data created in Test 3 should match this data.
gla11_reference_*.dat	GLA11 reference data created and verified during integration testing. The GLA11 data created in Test 3 should match this data.
gla12_reference_*.dat	GLA12 reference data created and verified during integration testing. The GLA12 data created in Test 4 should match this data.
gla13_reference_*.dat	GLA13 reference data created and verified during integration testing. The GLA13 data created in Test 4 should match this data.
gla14_reference_*.dat	GLA14 reference data created and verified during integration testing. The GLA14 data created in Test 4 should match this data.
gla15_reference_*.dat	GLA15 reference data created and verified during integration testing. The GLA15 data created in Test 4 should match this data.

tions of the GLA01 and GLA02 data are provided in the following subsections.

3.4.1 Detailed Description of GLA01 Input Data

The input GLA01 test data was based on 241 waveforms from 81 Bigfoot and 160 SLA waveforms, provided by the development team. For each surface type, (land or sea) the 241 waveforms are used sequentially and repeated as needed. The position is based on Schutz's 8-day ground track, where the 10 sec positions have been interpolated to 1/40 s.

There are approximately 25 minutes of data in first file, from equator to 50 deg N.

* Values may be different from the ones used during Waveform unit testing

```
Altimetry Record Index – created, starts from 1
Time of First Shot – created
Spares = 0
Delta Shot Times (39) – created
Predicted Lat (40) – calculated
Predicted Lon (40) – calculated
Spacecraft Height (40) – calculated
AD Gain Setting (40) – created
Filter Number (40) = 1
Ending Time of Range (40) – calculated from waveform
* Last Threshold Crossing Time (40) – calculated
* Next to Last Threshold Crossing (40) – calculated
* Echo Peak Location (40) – calculated
* Echo Peak Value (40)– calculated
Filter Weight Factors (40) – created
* Selected Filter Threshold (40) – calculated
Time of Transmit Pulse (40) – calculated
Starting Address Pulse Form (40) – created
Transmit Waveform Peak Status Flag (40) – created
1064 nm Laser Transmit Energy (40) – created
Range Window Status Word (40)= 0
Spares = 0
Waveform Compression Parameters: Large: 544,0; Small:200,0
Shot Counter (40) = 1 to 65500
Compression Ratios = 1,1
DEM Grid Number – calculated from internal DEM, GTOPO-30 5 km resolution
Spares = 0
Global DEM Bias = 100, 100
DEM Flag = 1
Filter Numbers Mask = 0
Spares = 0
Altimeter Digitizer Bias = 100, 100
Range Window Start and Stop = 100, 100
Sampled Transmit Pulse Waveform (40) – calculated
Spares = 0
* Background Mean Value (8 for short, 20 for long) – calculated
```

```
* Background Standard Deviation (8 for short, 20 for long)– calculated
1064 nm Range Waveform (8 for long, 20 for short) –calculated
Spares = 0
```

3.4.2 Detailed Description of GLA02 Input Data

The input GLA02 test data was based on data provided by the Atmosphere science team.

```
Atmosphere Record Index, i_rec_ndx –created
Time of First Sample, d_UTCTime –created
Predicted Latitude, d1_pred_lat –calculated
Predicted Longitude, d1_pred_lon –calculated
Spares –0
Location of Last Peak, i_loc_last_peak (40) –calculated
Etalon Filter Temperature, d_EtFiltTemp –calculated
Etalon Filter Feedback Monitor, d_EtFiltFBMon –calculated
Etalon Filter Heater Drive Control Point, d_EtFiltHtrCP –calculated
DEM Grid Number, i_DEM_grid_num –calculated
DEM Flag, i_DEM_flag  –calculated
532 nm LIDAR use flag, i1_g_lid_uf –calculated
532 nm LIDAR Data Quality Flag, i1_g_lid_qf –calculated
Spares –0
532 nm LIDAR Data 10 - -1 km, d40_g_lid –calculated
532 nm LIDAR Data 20 – 10 km, d5_g_lid –calculated
532 nm LIDAR 40 – 20 km, d1_g_lid –calculated
532 nm Saturation Flag 10 – 1 km, i40_g_sat_f –calculated
532 nm Saturation Flag 20 – 10 km, i5_g_sat_f –calculated
532 nm Saturation Flag 40  - 20 km, i1_g_sat_f –calculated
532 nm Laser Transmit Energy , d40_g_TxNrg_EU –calculated
532 nm Laser Transmit Energy, d5_g_TxNg_EU –calculated
532 nm Laser Transmit Energy, d1_g_TxNrg_EU –calculated
532 nm Integrated Return 40 – 20 km, d_ir_intRet –calculated
Start Range of Backscatter Profile, d_Rng2Prof –calculated
Range from S/C to Return Bin, d_rngSc2RetBin –calculated
Spares –0
Surface Elevation Peak of Return, d_SrfEl_PkRt –calculated
Surface Elevation Threshold of Return, d_SrfEl_ThRt –calculated
532 nm Background at 40 hz, d40_g_BG –calculated
532 nm Background at 5 hz, d5_g_BG –calculated
532 nm Background at 1 hz, d1_g_BG –calculated
```

532 nm Predicted Cloud Top Height, d_gPredCldTop –calculated
532 nm LIDAR Data Shot Counter, d1_g_shot_ctr – calculated
SPCM Background 2 and LIDAR Delay, d_SpcmBg2LdDel – calculated
SPCM Gate and Background 1 Delay, d_SpcmGatBg1Del –calculated
Photon Counter Range Delay, d_rdel_pc –calculated
SPCM Status, i_Spcm_stat –calculated
Spares –0
532 LIDAR Data Error Flag, i40_g_lid_ef –calculated
532 nm Laser Transmit Energy Count, d_TxNrg_Cts –calculated
532 nm Laser Transmit Energy Quality Flag, i_g_TxNrg_qf –calculated
Photon Counter Range Bias, i_pc_rbias –calculated
Integrated Return Quality Flag, i_ir_IntRet_qf –calculated
SPM Raw Counts, i_spcm_cts –calculated
1064 nmLIDAR Use Flag, i1_ir_lid_uf –calculated
1064 nm LIDAR Data Quality Flag, i1_ir_lid_qf –calculated
Spares –0
1064 nm Laser Transmit Energy, d40_ir_TxNrgEU –calculated
1064 nm Laser Transmit Energy, d5_ir_TxNrgEU –calculated
Spares –0
1064 nm Background –5 to –3.75 km at 40 hz, d40_ir_BG –calculated
1064 nm Background –5 to –3.75 km at 5 hz, d5_ir_BG –calculated
1064 nm LIDAR 10 - -1km, d40_ir_lid –calculated
1064 nm LIDAR 20 – 10 km, d5_ir_lid –calculated
1064 nm Cloud Digitizer Shot Counter, i_ir_shot_ctr – calculated
Cloud Digitizer Background 2 Delay, d_CdBg2Lid_Del –calculated
Cloud Digitizer Background 1 Delay, d_Cd_Bg1_Del –calculated
Range Gate for Cloud Digitizer, d_cd_rgate –calculated
Cloud Digitizer Detector Status, d_cd_det_stat –calculated
Spares –0
1064 nm Cloud Digitizer Data Spare, d40_ir_Spare –calculated
Cloud Return Peak Signal, d_CldPkSig –calculated
A/D Output Attenuation Setting, d_cd_ad_att –calculated
Cloud Digitizer Range Bias, d_cd_rbias –calculated
1064 Digitizer Gain Setting, d_ir_gainSet –calculated
Threshold Crossing, d_thres_xing –calculated
Ground Return Peak Signal, d_gndret_pksig –calculated
Ground Return Location, d_gnd_ret_loc –calculated
1064 nm Laser Transmit Energy Quality Flag, i_ir_TxNrg_qf –calculated

```
Spares –0
```

Section 4

Procedures

4.1 Environment

The GSAS software has been written and tested under the HP / UX 9.0.4 / Fortran 90 1.0 and HP / UX 11.0 / Fortran 90 v2.3 environments. It is recommended that the acceptance test be run in an environment identical one of those used for development and testing. The team has identified problems with other versions of the HP compiler, so the version of the compiler is critical.

The team recommends, at a minimum, an HP L-class server with 1GB of RAM and 1GB of free disk space.

4.2 Software Installation

Software should be installed and compiled as described in the GSAS User's Guide. This document will use the same conventions as the User's Guide with respect to directory designations. $GLAS_HOME will be designated as the directory in which the software was installed.

Integration testing was performed with no make options (neither DEBUG nor FAST were used). Acceptance testing should be performed with the same compiler flags.

4.3 Test Directory Setup

After the software has been installed, change to the $GLAS_HOME / test directory.

```
cd $GLAS_HOME/test
```

Run the setup_acceptance_test.sh script.

```
./setup_acc_test.sh
```

This script will create and populate five test directories, one for each of the five test cases. The contents of the test directories are identified in Section 3.4.

4.4 Recovery Steps

If a test procedure fails unexpectedly, the tester should look in the ANC06 file for error messages. Output products may also contain information as to the problem. Also verify the operating system and compiler versions.

Reference products should be verified using the UNIX 'cksum' command and the results compared with the checksum references (`$GLAS_HOME/test/test_data/00checksums` file). Checksum differences mean that the reference test data have been corrupted and that the distribution should be reinstalled.

Before re-running a test, the tester must remove or rename all output files created during the prior run. Scripts named “cleanup.sh’ have been provided in each test directory to do this.

If input files are accidently deleted, re-run the setup_acceptance_test.sh script to return to the default state.

If files were unrecoverably deleted, reinstall the software from the distribution media.

4.5 Test Cases

This section describes the five test cases of the GSAS Acceptance Test. See Appendix A of the GSAS User’s Guide to identify input and output file types. Additional debug input/outputs may be created in each test if differences are found in the output binary files. The debug files are described in the Product Reader section of the GSAS User’s Guide. <u>Tests 4 and 5 must be run only after successful completion of Test 2 since the output of Test 2 serves as the input to Tests 4 and 5.</u>

4.5.1 Test 1: Level 1A Processing

This test uses 90 seconds of known GLA00 data to create multi-granule GLA01-03. It tests the functionallity of reading GLA00 APIDs and writing multiple granules of GLA01-03. The input GLA00 data is only bitwise valid, i.e. if a field should contain a 2 byte integer, then it did, but the value of the integer had nothing to do with what it should be from real telemetry. The GLA01,02, and 03 files created from this test are totally unusable for further processing since the parameters do not contain reasonable values.

All input files will have been linked into the test directory by the shell script run in Section 4.3. This test case takes less than a minute to run on an HP L-class server.

4.5.1.1 Inputs

```
cf01_001_20001027_001.ctl
anc07_001_01_00.dat
anc07_001_01_01.dat
anc07_001_01_05.dat
anc20_001_20000419_020725_00.dat
gla00_001_20000101_000001_01_00.dat
gla00_001_20000101_000021_01_00.dat
gla00_001_20000101_000041_01_00.dat
```

4.5.1.2 Outputs

```
anc06_001_20001026_001.dat
gla01_001_11_001_0001_1_01_00.dat
gla01_001_11_001_0001_2_01_00.dat
gla01_001_11_001_0001_3_01_00.dat
gla02_001_11_001_0001_1_01_00.dat
gla02_001_11_001_0001_2_01_00.dat
gla02_001_11_001_0001_3_01_00.dat
gla03_001_11_001_0001_1_01_00.dat
```

```
gla03_001_11_001_0001_2_01_00.dat
gla03_001_11_001_0001_3_01_00.dat
```

4.5.1.3 Procedure

Change to the $GLAS_HOME/test/test1 directory. Verify that the listed input files exist and that the listed output files do not exist. If necessary, delete or rename old output files. Run GLAS_Exec with the control file provided.

```
cd $GLAS_HOME/test/test1
setenv SHLIB_PATH.
./GLAS_Exec cf01_001_20001027_001.ctl
```

Run the verify.sh script. This script will compare checksums of the output products with previously-tested (and verified) products checksums. If differences are found, the script will automatically run the appropriate product reader to create human-readable output files. The reader output files will be differenced (using the diff command) with the outputs from the appropriate reference file.

4.5.2 Test 2: Waveform Processing

This test uses one hour of known GLA01 data to create multi-granule GLA05. The GLA01 data were created by the science team from shuttle (SLA02) and aircraft (bigfoot) altimetry. The latitude and longitude on the input data was from the nominal 8-day reference orbit received from UTCSR. 56 granules of GLA01 data were created covering the first 14 revolutions of the reference orbit. The first granule which was used for this test was only 13 minutes long because it started at the equator, the 2nd and 3rd granules were approximately 25 minutes each. The team arbitrarily repeated the 81 bigfoot and 160 SLA02 waveforms throughout the granules without making any attempt to correlate the actual aircraft groundtrack location with the GLAS groundtrack on GLA01. Therefore one cannot use this file or GLA05 to infer waveform characteristics for specific locations. Short records were created when the GLAS groundtrack was determined to be over ocean, based on the DEM and long records were created when it was determined to be over land. This test exercised the waveform fitting algorithms for the transmit and the received pulse, the mechanics of the calculations of the range measurements from the waveform positions, the skewness and kurtosis calculations, and the calculation of location using the precision orbit, attitude, and ITRF rotation matrix files. The waveforms themselves, were not truly typical of GLAS waveforms, but similar. Therefore one can see from GLA05 how the GLAS algorithms would process these waveforms, but cannot infer whether the algorithms are optimal for GLAS waveforms. The resolution of the received waveforms was different than that of GLAS, however the transmitted waveform was consistent with GLAS theoretically. Therefore the received waveforms may be narrower than the transmitted pulse which is not physically possible.

All input files will have been linked into the test directory by the shell script run in Section 4.3. This test case takes approximately 1.5 hours to run on an HP L-class server.

4.5.2.1 Inputs

```
cf02_001_20001027_001.ctl
anc07_001_01_00.dat
anc07_001_01_01.dat
anc07_001_01_03.dat
anc07_001_01_04.dat
anc08_001_20000101_000000_01_00.dat
anc09_001_20000101_000000_01_00.dat
anc12_001_01_00.dat
anc12_001_01_01.dat
anc24_001_20000101_120000_01_00.dat
gla01_001_11_001_0001_1_01_00.dat
gla01_001_11_001_0001_2_01_00.dat
gla01_001_11_001_0001_3_01_00.dat
```

4.5.2.2 Outputs

```
anc06_001_20001027_001.dat
gla05_001_11_001_0001_1_01_00.dat
gla05_001_11_001_0001_2_01_00.dat
gla05_001_11_001_0001_3_01_00.dat
qap05_001_11_001_0001_1_01_00.dat
```

4.5.2.3 Procedure

Change to the $GLAS_HOME/test/test2 directory. Verify that the listed input files exist and that the listed output files do not exist. If necessary, delete or rename old output files. Run GLAS_Exec with the control file provided. This process is computation-intensive and make take a while to run.

```
cd $GLAS_HOME/test/test2
./GLAS_Exec cf02_001_20001027_001.ctl
```

Run the verify.sh script. This script will compare checksums of the output products with previously-tested (and verified) products checksums. If differences are found, the script will automatically run the appropriate product reader to create human-readable output files. The reader output files will be differenced (using the diff command) with the outputs from the appropriate reference file.

4.5.3 Test 3: Atmosphere Processing

This test uses known GLA02 data to create multi-granule GLA07-11. The input data were created by the lidar science team from ER-2 airborne lidar. The latitude and longitude on the input data are from the nominal 8-day reference orbit received from UTCSR. The first granule used for this test is only 13 minutes long because it started at the equator, the 2nd and 3rd granules are approximately 25 minutes each. The 446 seconds of lidar data were repeated throughout the granules without making any attempt to correlate the actual aircraft groundtrack location with the GLAS groundtrack. Therefore, one cannot use this file or GLA07-11 to infer atmospheric characteristics for specific locations. Also, the DEM was set to zero for every geographic point. This test exercises the cloud and aerosol detection and optical properties algorithms for the vertical profiles along the track. The data on GLA07-9 are valid physical results for the input data set. However, the data on GLA10-11, while self-consistent, are not physically valid for the data in GLA07-9 because the geographic locations were arbitrarily set.

All input files will have been linked into the test directory by the shell script run in Section 4.3. This test case takes approximately 12 minutes to run on an HP L-class server.

4.5.3.1 Inputs

```
cf03_001_20001030_001.ctl
anc01_001_20000101_120000_01_01.dat
anc01_001_20000101_120000_01_02.dat
anc01_001_20000101_120000_01_03.dat
anc01_001_20000101_120000_01_04.dat
anc01_001_20000101_180000_01_00.dat
anc01_001_20000101_180000_01_01.dat
anc01_001_20000101_180000_01_02.dat
anc01_001_20000101_180000_01_03.dat
anc01_001_20000101_180000_01_04.dat
anc07_001_01_00.dat
anc07_001_01_01.dat
anc07_001_01_02.dat
anc07_001_01_03.dat
anc08_001_20000101_000000_01_00.dat
anc09_001_20000101_000000_01_00.dat
anc12_001_01_00.dat
anc12_001_01_01.dat
anc18_001_01_00.dat
anc24_001_20000101_120000_01_00.dat
anc30_001_01_00.dat
anc31_001_01_00.dat
gla02_001_11_001_0001_1_00.dat
gla02_001_11_001_0001_2_00.dat
gla02_001_11_001_0001_3_00.dat
```

4.5.3.2 Outputs

```
anc06_001_20001030_001.dat
gla07_001_11_001_0001_1_00.dat
gla08_001_11_001_0001_1_00.dat
gla09_001_11_001_0001_1_00.dat
gla10_001_11_001_0001_1_00.dat
gla11_001_11_001_0001_1_00.dat
gla07_001_11_001_0001_2_00.dat
gla08_001_11_001_0001_2_00.dat
gla09_001_11_001_0001_2_00.dat
gla10_001_11_001_0001_2_00.dat
gla11_001_11_001_0001_2_00.dat
gla07_001_11_001_0001_3_00.dat
gla08_001_11_001_0001_3_00.dat
gla09_001_11_001_0001_3_00.dat
gla10_001_11_001_0001_3_00.dat
gla11_001_11_001_0001_3_00.dat
```

4.5.3.3 Procedure

Change to the $GLAS_HOME/test/test3 directory. Verify that the listed input files exist and that the listed output files do not exist. If necessary, delete or rename old output files. Run GLAS_Exec with the control file provided.

```
cd $GLAS_HOME/test/test3
./GLAS_Exec cf03_001_20001030_001.ctl
```

Run the verify.sh script. This script will compare checksums of the output products with previously-tested (and verified) products checksums. If differences are found, the script will automatically run the appropriate product reader to create human-readable output files. The reader output files will be differenced (using the diff command) with the outputs from the appropriate reference file.

4.5.4 Test 4: Elevation Processing

This test uses the GLA05 data created in Test 2 to create multi-granule GLA06,12-15. This test confirms the mechanics of calculating the individual elevations from GLA05 input for GLA06 and each of the regional level 2 products. The rules used for each of the regional products to define location and elevation were tested as defined in the ATBD. However the regional masks were not defined so each regional level 2 product contained all of the GLA06 data instead of just data within it's specific mask.

This test also exercises the geoid interpolation algorithm, the tide calculation code, and the atmospheric corrections (wet and dry troposphere) code. The inputs to all these algorithms are reasonable, and in the correct format, but not necessarily the final input to be used in mission operations. The geoid is the EGM96 geoid. The tide coefficients are from a reasonable model, but not the one selected for GLAS operations. The meteorological data used is from the same source that will be used for GLAS, but was for July of 1999 whereas the data was marked as being for January 2000 so the answers would be for the wrong season.

All input files will have been linked into the test directory by the shell script run in Section 4.3. This test can only be performed if Test 2 passed acceptance testing and the resultant output files were not removed.

This test case takes approximately 5 minutes to run on an HP L-class server.

4.5.4.1 Inputs

```
cf04_001_20001027_001.ctl
anc01_001_20000101_120000_01_01.dat
anc01_001_20000101_120000_01_02.dat
anc01_001_20000101_120000_01_03.dat
anc01_001_20000101_120000_01_04.dat
anc01_001_20000101_180000_01_00.dat
anc01_001_20000101_180000_01_01.dat
anc01_001_20000101_180000_01_02.dat
anc01_001_20000101_180000_01_03.dat
anc01_001_20000101_180000_01_04.dat
anc07_001_01_00.dat
anc07_001_01_01.dat
```

```
anc07_001_01_03.dat
anc08_001_20000101_000000_01_00.dat
anc09_001_20000101_000000_01_00.dat
anc12_001_01_00.dat
anc12_001_01_01.dat
anc13_001_01_00.dat
anc16_001_01_00.dat
anc17_001_01_00.dat
anc18_001_01_00.dat
anc24_001_20000101_120000_01_00.dat
anc30_001_01_00.dat
anc31_001_01_00.dat
gla05_001_11_001_0001_1_01_00.dat
gla05_001_11_001_0001_2_01_00.dat
gla05_001_11_001_0001_3_01_00.dat
```

4.5.4.2 Outputs

```
anc06_001_20001027_004.dat
gla06_001_11_001_0001_1_01_00.dat
gla06_001_11_001_0001_2_01_00.dat
gla06_001_11_001_0001_3_01_00.dat
gla12_001_11_001_0001_1_01_00.dat
gla12_001_11_001_0001_2_01_00.dat
gla12_001_11_001_0001_3_01_00.dat
gla13_001_11_001_0001_1_01_00.dat
gla13_001_11_001_0001_2_01_00.dat
gla13_001_11_001_0001_3_01_00.dat
gla14_001_11_001_0001_1_01_00.dat
gla14_001_11_001_0001_2_01_00.dat
gla14_001_11_001_0001_3_01_00.dat
gla15_001_11_001_0001_1_01_00.dat
gla15_001_11_001_0001_2_01_00.dat
gla15_001_11_001_0001_3_01_00.dat
```

4.5.4.3 Procedure

Change to the $GLAS_HOME/test/test4 directory. Verify that the listed input files exist and that the listed output files do not exist. If necessary, delete or rename old output files. Run GLAS_Exec with the control file provided.

```
cd $GLAS_HOME/test/test4
./GLAS_Exec cf04_001_20001027_001.ctl
```

Run the verify.sh script. This script will compare checksums of the output products with previously-tested (and verified) products checksums. If differences are found, the script will automatically run the appropriate product reader to create human-readable output files. The reader output files will be differenced (using the diff command) with the outputs from the appropriate reference file.

4.5.5 Test 5: Partial Elevation Processing

This test uses the GLA05 data created in Test 2 to create multi-granule GLA06 only. All input files will have been linked into the test directory by the shell script run in Section 4.3.

This test can only be performed if Test 2 passed acceptance testing and the resultant output files were not removed.

This test case takes approximately 5 minutes to run on an HP L-class server.

4.5.5.1 Inputs

```
cf05_001_20001027_001.ctl
anc01_001_20000101_120000_01_01.dat
anc01_001_20000101_120000_01_02.dat
anc01_001_20000101_120000_01_03.dat
anc01_001_20000101_120000_01_04.dat
anc01_001_20000101_180000_01_00.dat
anc01_001_20000101_180000_01_01.dat
anc01_001_20000101_180000_01_02.dat
anc01_001_20000101_180000_01_03.dat
anc01_001_20000101_180000_01_04.dat
anc07_001_01_00.dat
anc07_001_01_01.dat
anc07_001_01_03.dat
anc08_001_20000101_000000_01_00.dat
anc09_001_20000101_000000_01_00.dat
anc12_001_01_00.dat
anc12_001_01_01.dat
anc13_001_01_00.dat
anc16_001_01_00.dat
anc17_001_01_00.dat
anc18_001_01_00.dat
anc24_001_20000101_120000_01_00.dat
anc30_001_01_00.dat
anc31_001_01_00.dat
gla05_001_11_001_0001_1_01_00.dat
gla05_001_11_001_0001_2_01_00.dat
gla05_001_11_001_0001_3_01_00.dat
```

4.5.5.2 Outputs

```
anc06_001_20001027_005.dat
gla06_001_11_001_0001_1_01_00.dat
gla06_001_11_001_0001_2_01_00.dat
gla06_001_11_001_0001_3_01_00.dat
```

4.5.5.3 Procedure

Change to the $GLAS_HOME/test/test5 directory. Verify that the listed input files exist and that the listed output files do not exist. If necessary, delete or rename old output files. Run GLAS_Exec with the control file provided.

```
cd $GLAS_HOME/test/test5
./GLAS_Exec cf05_001_20001027_001.ctl
```

Run the verify.sh script. This script will compare checksums of the output products with previously-tested (and verified) products checksums. If differences are found, the script will automatically run the appropriate product reader to create human-readable output files. The reader output files will be differenced (using the diff command) with the outputs from the appropriate reference file.

Section 5

Evaluation Criteria

5.1 Test 1: Level 1A Processing

To pass acceptance testing, all tests performed by the verify.sh script must complete without failure. Successful results indicate that there are no differences between the products created during acceptance testing and those verified during integration testing. A waiver may obtained to explain differences caused by operating system, compiler versions or optimization levels.

Known Issues:

> Team-generated test data is used because of structure alignment difficulties with the existing GLA00 APID structures. Additionally, the test data is not scientifically realistic and cannot be used for further Waveform, Atmosphere, or Elevation processing.

5.2 Test 2: Waveform Processing

To pass acceptance testing, all tests performed by the verify.sh script must complete without failure. Successful results indicate that there are no differences between the products created during acceptance testing and those verified during integration testing. A waiver may obtained to explain differences caused by operating system, compiler versions or optimization levels.

Known Issues:

> GLAS_Exec reports the following error: "...-30008, 1, WFMgr, Error from WF_Mgr Filter index is invalid, calculating noise, sDevNs and maxFilterW for WF...". This is caused by an inconsistency in the test data and should be ignored.

5.3 Test 3: Atmosphere Processing

To pass acceptance testing, all tests performed by the verify.sh script must complete without failure. Successful results indicate that there are no differences between the products created during acceptance testing and those verified during integration testing. A waiver may obtained to explain differences caused by operating system, compiler versions or optimization levels.

Known Issues:

> The Atmosphere subsystem does not handle all situations correctly with regards to invalid data. Originally, the Atmosphere integration and acceptance test used multi-granules files matching the GLA05/06 timespans.
>
> GLAS_Exec reports the several errors in the ANC06 file. These is caused by an inconsistency in the test data and should be ignored. A listing of the ANC06 error summary is provided below as an example:

```
ERROR_SUMMARY = 51544  315 Ratio of integrated returns out-of-bounds
ERROR_SUMMARY = 51544  2026 Divide by zero
ERROR_SUMMARY = 51544  14 Exponent too large
ERROR_SUMMARY = 51544  943 Input PBL height out-of-bounds
ERROR_SUMMARY = 51544  52 Insufficient filtered data for processing
ERROR_SUMMARY = 51544  800 Random noise not valid for cloud detection
```

5.4 Test 4: Elevation Processing

To pass acceptance testing, all tests performed by the verify.sh script must complete without failure. Successful results indicate that there are no differences between the products created during acceptance testing and those verified during integration testing. A waiver may obtained to explain differences caused by operating system, compiler versions or optimization levels.

GLAS_Exec reports the following error: “Bad Frame flag set in E_IceSheetParm”. This is caused by an inconsistency in the test data and should be ignored.

5.5 Test 5: Partial Elevation Processing

To pass acceptance testing, all tests performed by the verify.sh script must complete without failure. Successful results indicate that there are no differences between the products created during acceptance testing and those verified during integration testing. A waiver may obtained to explain differences caused by operating system, compiler versions or optimization levels.

GLAS_Exec reports the following error: “Bad Frame flag set in E_IceSheetParm”. This is caused by an inconsistency in the test data and should be ignored.

Section 6

Expected Results

6.1 Test 1: Level 1A Processing

Output of a successful acceptance test is shown below:

```
Verifying Test 1 outputs....


Passed: gla01_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla01_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla01_001_11_001_0001_3_01_00.dat matches the reference

Passed: gla02_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla02_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla02_001_11_001_0001_3_01_00.dat matches the reference

Passed: gla03_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla03_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla03_001_11_001_0001_3_01_00.dat matches the reference

Done.
```

6.2 Test 2: Waveform Processing

Output of a successful acceptance test is shown below:

```
Verifying Test 2 outputs....


Passed: gla05_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla05_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla05_001_11_001_0001_3_01_00.dat matches the reference

Done.
```

6.3 Test 3: Atmosphere Processing

Output of a successful acceptance test is shown below:

```
Verifying Test 3 outputs....
```

```
Passed: gla07_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla07_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla07_001_11_001_0001_3_01_00.dat matches the reference

Passed: gla08_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla08_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla08_001_11_001_0001_3_01_00.dat matches the reference

Passed: gla09_001_11_001_0001_1_01_00.dat matches the reference.

Passed: gla09_001_11_001_0001_2_01_00.dat matches the reference.

Passed: gla09_001_11_001_0001_3_01_00.dat matches the reference.

Passed: gla10_001_11_001_0001_1_01_00.dat matches the reference.

Passed: gla10_001_11_001_0001_1_01_00.dat matches the reference.

Passed: gla10_001_11_001_0001_1_01_00.dat matches the reference.

Passed: gla11_001_11_001_0001_1_01_00.dat matches the reference.

Passed: gla11_001_11_001_0001_2_01_00.dat matches the reference.

Passed: gla11_001_11_001_0001_3_01_00.dat matches the reference.

Done.
```

6.4 Test 4: Elevation Processing

Output of a successful acceptance test is shown below:

```
Verifying Test 4 outputs....


Passed: gla06_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla06_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla06_001_11_001_0001_3_01_00.dat matches the reference

Passed: gla12_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla12_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla12_001_11_001_0001_3_01_00.dat matches the reference

Passed: gla13_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla13_001_11_001_0001_2_01_00.dat matches the reference
```

```
Passed: gla13_001_11_001_0001_3_01_00.dat matches the reference

Passed: gla14_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla14_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla14_001_11_001_0001_3_01_00.dat matches the reference

Passed: gla15_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla15_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla15_001_11_001_0001_3_01_00.dat matches the reference

Done.
```

6.5 Test 5: Elevation Partial Processing

Output of a successful acceptance test is shown below:

```
Verifying Test 5 outputs....

Passed: gla06_001_11_001_0001_1_01_00.dat matches the reference

Passed: gla06_001_11_001_0001_2_01_00.dat matches the reference

Passed: gla06_001_11_001_0001_3_01_00.dat matches the reference

Done.
```

Section 7

Actual Results

7.1 Test 1: Level 1A Processing

Passed. Acceptance test outputs exactly matched the integration test references.

7.2 Test 2: Waveform Processing

Passed. Acceptance test outputs exactly matched the integration test references.

7.3 Test 3: Atmosphere Processing

Passed. Acceptance test outputs exactly matched the integration test references.

7.4 Test 4: Elevation Processing

Passed. Acceptance test outputs exactly matched the integration test references.

7.5 Test 5: Elevation Partial Processing

Passed. Acceptance test outputs exactly matched the integration test references.

Appendix A
GSAS V1 Acceptance Test Control Files

This Appendix lists the GSAS V1 Acceptance Test control files.

A.1 Test 1 Control File - L1A

```
#
#----------Start of Control File
#
# This is the control file for the L1A acceptance test.
#
#----------Execution Information
#
= GLAS_Exec
TEMPLATE=l1a_control_file
EXEC_KEY=10023425
DATE_GENERATED=2000 October 27
OPERATOR=jlee
#
#----------Cycle Track Boundaries
#
# Cycle/Track Boundaries
#
CYCLE=001 0000000 0000060
TRACK=001 0000000 0000060
#
#----------Static ANC Files
#
# Input ANC07 Files : 00=error, 01=global, 05=L1A
#
INPUT_FILE=anc07_001_01_00.dat 0000000 0000060
INPUT_FILE=anc07_001_01_01.dat 0000000 0000060
INPUT_FILE=anc07_001_01_05.dat 0000000 0000060
#
#----------Dynamic ANC Files
#
# POD (Predicted)- Stop time MUST be > stop time of data !!!
#
INPUT_FILE=anc20_001_20000101_000000_01_00.dat 00000 0000061
#
#----------Input GLA Files
#
# Input GLA00 Files
#
INPUT_FILE=gla00_001_20000101_000001_01_00.dat 000001 0000020
INPUT_FILE=gla00_001_20000101_000021_01_00.dat 0000021 0000040
INPUT_FILE=gla00_001_20000101_000041_01_00.dat 0000041 0000060
#
#----------Output ANC Files
#
```

```
# Output ANC06
#
OUTPUT_FILE=anc06_001_20001026_001.dat 0000000 0000060
#
#----------Output GLA Files
#
# Output GLA01 Files
#
OUTPUT_FILE=gla01_001_11_001_0001_1_01_00.dat 0000001 0000020
OUTPUT_FILE=gla01_001_11_001_0001_2_01_00.dat 0000021 0000040
OUTPUT_FILE=gla01_001_11_001_0001_3_01_00.dat 0000041 0000060
#
# Output GLA02 Files
#
OUTPUT_FILE=gla02_001_11_001_0001_1_01_00.dat 0000001 0000020
OUTPUT_FILE=gla02_001_11_001_0001_2_01_00.dat 0000021 0000040
OUTPUT_FILE=gla02_001_11_001_0001_3_01_00.dat 0000041 0000060
#
# Output GLA03 Files
#
OUTPUT_FILE=gla03_001_11_001_0001_1_01_00.dat 0000001 0000020
OUTPUT_FILE=gla03_001_11_001_0001_2_01_00.dat 0000021 0000040
OUTPUT_FILE=gla03_001_11_001_0001_3_01_00.dat 0000041 0000060
#
#----------Execution Control
#
L1A_PROCESS=ALL
#
#----------End of Control File
#
=New Utility
```

A.2 Test 2 Control File - Waveforms

```
#
#----------Start of Control File
#
# The control file for the Waveform acceptance test
#
#----------Execution Information
#
= GLAS_Exec
TEMPLATE=wf_control_file
EXEC_KEY=10023425
DATE_GENERATED=2000 October 27
OPERATOR=jlee
#
#----------Cycle Track Boundaries
#
# Cycle/Track Boundaries
#
CYCLE=001 0000000 0000060
TRACK=0001 0000001 0004998
```

```
#
#----------Static ANC Files
#
# Input ANC07 Files : 00=error, 01=global, 03=elevation 04=waveform
# ** Elevation is needed because of PR20001031-001
#
INPUT_FILE=anc07_001_01_00.dat 0000000 0050006
INPUT_FILE=anc07_001_01_01.dat 0000000 0050006
INPUT_FILE=anc07_001_01_03.dat 0000000 0050006
INPUT_FILE=anc07_001_01_04.dat 0000000 0050006
#
# Input DEM Files : 00 = DEM, 01=Mask
#
INPUT_FILE=anc12_001_01_00.dat 0000000 0050006
INPUT_FILE=anc12_001_01_01.dat 0000000 0050006
#
# Input Standard Atmosphere File
#
INPUT_FILE=anc18_001_01_00.dat 0000000 0050006
#
#----------Dynamic ANC Files
#
# Input MET Files : 00 = hdr, 01=pwat, 02=hgt, 03=rh, 04=tmp
#
# 1st 6 hours
#
INPUT_FILE=anc01_001_20000101_120000_01_00.dat 0000000 0021599
INPUT_FILE=anc01_001_20000101_120000_01_01.dat 0000000 0021599
INPUT_FILE=anc01_001_20000101_120000_01_02.dat 0000000 0021599
INPUT_FILE=anc01_001_20000101_120000_01_03.dat 0000000 0021599
INPUT_FILE=anc01_001_20000101_120000_01_04.dat 0000000 0021599
#
# 2nd 6 hours
#
INPUT_FILE=anc01_001_20000101_180000_01_00.dat 0021600 0043199
INPUT_FILE=anc01_001_20000101_180000_01_01.dat 0021600 0043199
INPUT_FILE=anc01_001_20000101_180000_01_02.dat 0021600 0043199
INPUT_FILE=anc01_001_20000101_180000_01_03.dat 0021600 0043199
INPUT_FILE=anc01_001_20000101_180000_01_04.dat 0021600 0043199
#
# POD (precision) - Stop time MUST be > stop time of data !!!
#
INPUT_FILE=anc08_001_20000101_000000_01_00.dat 0000000 0050006
#
# PAD
#
INPUT_FILE=anc09_001_20000101_000000_01_00.dat 0000000 0050006
#
# Rotation Matrix
#
INPUT_FILE=anc24_001_20000101_120000_01_00.dat 0000000 0050006
#
#----------Input GLA Files
#
```

```
# Input GLA01 Files
#
INPUT_FILE=gla01_001_11_001_0001_1_01_00.dat 0000001 0000808
INPUT_FILE=gla01_001_11_001_0001_2_01_00.dat 0000808 0002100
INPUT_FILE=gla01_001_11_001_0001_3_01_00.dat 0002100 0003711
#
#----------Output ANC Files
#
# Output ANC06
#
OUTPUT_FILE=anc06_001_20001027_001.dat 0000000 0081201
#
#----------Output GLA Files
#
# Output GLA05 Files
#
OUTPUT_FILE=gla05_001_11_001_0001_1_01_00.dat 0000001 0000808
OUTPUT_FILE=gla05_001_11_001_0001_2_01_00.dat 0000808 0002100
OUTPUT_FILE=gla05_001_11_001_0001_3_01_00.dat 0002100 0003711
#
#----------Output QAP Files
#
# QAP files are not required for v1, thus we will use only one file
# for the whole run. In production, there should be one QAP file
# for each GLA file and the naming convention will follow the GLA
# naming convention.
#
OUTPUT_FILE=qap05_001_11_001_0001_1_01_00.dat 0000000 0050006
#
#----------Execution Control
#
WAVEFORM_PROCESS=ALL
#
#----------End of Control File
#
```

A.3 Test 3 Control File - Atmosphere

```
#
#----------Start of Control File
#
# The control file for the Atmosphere acceptance test
#
#----------Execution Information
#
= GLAS_Exec
TEMPLATE=atm_control_file
EXEC_KEY=10023429
DATE_GENERATED=2000 October 30
OPERATOR=jlee
#
#----------Cycle Track Boundaries
#
```

```
# Cycle/Track Boundaries
#
CYCLE=001 0000000 0000060
TRACK=0001 0000001 0004998
#
#----------Static ANC Files
#
# Input ANC07 Files : 00=error, 01=global, 02=atmosphere, 03=elevation
#
INPUT_FILE=anc07_001_01_00.dat 0000000 0050006
INPUT_FILE=anc07_001_01_01.dat 0000000 0050006
INPUT_FILE=anc07_001_01_02.dat 0000000 0050006
INPUT_FILE=anc07_001_01_03.dat 0000000 0050006
#
# Input DEM Files : 00 = DEM, 01=Mask
#
INPUT_FILE=anc12_001_01_00.dat 0000000 0050006
INPUT_FILE=anc12_001_01_01.dat 0000000 0050006
#
# Input Standard Atmosphere File
#
INPUT_FILE=anc18_001_01_00.dat 0000000 0050006
#
# Input Trop and Aerosol map Files
#
INPUT_FILE=anc30_001_01_00.dat 0000000 0050006
INPUT_FILE=anc31_001_01_00.dat 0000000 0050006
#
#----------Dynamic ANC Files
#
# Input MET Files : 00 = hdr, 01=pwat, 02=hgt, 03=rh, 04=tmp
#
# 1st 6 hours
#
INPUT_FILE=anc01_001_20000101_120000_01_00.dat 0000000 0021599
INPUT_FILE=anc01_001_20000101_120000_01_01.dat 0000000 0021599
INPUT_FILE=anc01_001_20000101_120000_01_02.dat 0000000 0021599
INPUT_FILE=anc01_001_20000101_120000_01_03.dat 0000000 0021599
INPUT_FILE=anc01_001_20000101_120000_01_04.dat 0000000 0021599
#
# 2nd 6 hours
#
INPUT_FILE=anc01_001_20000101_180000_01_00.dat 0021600 0043199
INPUT_FILE=anc01_001_20000101_180000_01_01.dat 0021600 0043199
INPUT_FILE=anc01_001_20000101_180000_01_02.dat 0021600 0043199
INPUT_FILE=anc01_001_20000101_180000_01_03.dat 0021600 0043199
INPUT_FILE=anc01_001_20000101_180000_01_04.dat 0021600 0043199
#
# POD (precision) - Stop time MUST be > stop time of data !!!
#
INPUT_FILE=anc08_001_20000101_000000_01_00.dat 0000000 0050006
#
# PAD
#
```

```
INPUT_FILE=anc09_001_20000101_000000_01_00.dat 0000000 0050006
#
# Rotation Matrix
#
INPUT_FILE=anc24_001_20000101_120000_01_00.dat 0000000 0050006
#
#----------Input gla Files
#
# Input GLA02 Files
#
INPUT_FILE=gla02_001_11_001_0001_1_01_00.dat 0000001 0000808
#INPUT_FILE=gla02_001_11_001_0001_2_01_00.dat 0000808 0002100
#INPUT_FILE=gla02_001_11_001_0001_3_01_00.dat 0002100 0003711
#
#----------Output ANC Files
#
# Output ANC06
#
OUTPUT_FILE=anc06_001_20001030_001.dat 0000000 0050006
#
#----------Output GLA Files
#
# Output GLA07 Files
#
OUTPUT_FILE=gla07_001_11_001_0001_1_01_00.dat 0000001 0000808
#OUTPUT_FILE=gla07_001_11_001_0001_2_01_00.dat 0000808 0002100
#OUTPUT_FILE=gla07_001_11_001_0001_3_01_00.dat 0002100 0003711
#
# In production, GLA08-11 are not a 1-1 match with GLA06. The
# original integration tests were run this way, so the acceptance
# test will be this way as well. Normally, GLA08-12 are
# not segment-dependent.
#
# Output GLA08
#
OUTPUT_FILE=gla08_001_11_001_0001_1_01_00.dat 0000001 0000808
#OUTPUT_FILE=gla08_001_11_001_0001_2_01_00.dat 0000808 0002100
#OUTPUT_FILE=gla08_001_11_001_0001_3_01_00.dat 0002100 0003711
#
# Output GLA09
#
OUTPUT_FILE=gla09_001_11_001_0001_1_01_00.dat 0000001 0000808
#OUTPUT_FILE=gla09_001_11_001_0001_2_01_00.dat 0000808 0002100
#OUTPUT_FILE=gla09_001_11_001_0001_3_01_00.dat 0002100 0003711
#
# Output GLA10
#
OUTPUT_FILE=gla10_001_11_001_0001_1_01_00.dat 0000001 0000808
#OUTPUT_FILE=gla10_001_11_001_0001_2_01_00.dat 0000808 0002100
#OUTPUT_FILE=gla10_001_11_001_0001_3_01_00.dat 0002100 0003711
#
# Output GLA11
#
OUTPUT_FILE=gla11_001_11_001_0001_1_01_00.dat 0000001 0000808
```

```
#OUTPUT_FILE=gla11_001_11_001_0001_2_01_00.dat 0000808 0002100
#OUTPUT_FILE=gla11_001_11_001_0001_3_01_00.dat 0002100 0003711
#
#----------Output QAP Files
#
# QAP files are not required for v1, thus we will use only one file
# for the whole run. In production, there should be one QAP file
# for each GLA file and the naming convention will follow the GLA
# naming convention.
#
# Output QAP07 Files
#
OUTPUT_FILE=qap07_001_11_001_0001_1_00.dat 0000000 0050006
#
# Output QAP08-11
#
OUTPUT_FILE= qap08_001_11_001_0001_1_00.dat 0000000 0050006
#
#----------Execution Control
#
ATMOSPHERE_PROCESS=ALL
#
#----------End of Control File
#
=New Utility

#
```

A.4 Test 4 Control File - Elevation

```
#
#----------Start of Control File
#
# The control file for the Elevation acceptance test
#
#----------Execution Information
#
= GLAS_Exec
TEMPLATE=elev_control_file
EXEC_KEY=10023426
DATE_GENERATED=2000 October 27
OPERATOR=jlee
#
#----------Cycle Track Boundaries
#
# Cycle/Track Boundaries
#
CYCLE=001 0000000 0000060
TRACK=0001 0000001 0004998
TRACK=0002 0004998 0010798
#
#----------Static ANC Files
#
# Input ANC07 Files : 00=error, 01=global, 03=elevation
```

```
#
INPUT_FILE=anc07_001_01_00.dat  0000000 0050006
INPUT_FILE=anc07_001_01_01.dat  0000000 0050006
INPUT_FILE=anc07_001_01_03.dat  0000000 0050006
#
# Input DEM Files : 00 = DEM, 01=Mask
#
INPUT_FILE=anc12_001_01_00.dat  0000000 0050006
INPUT_FILE=anc12_001_01_01.dat  0000000 0050006
#
# Input Standard Atmosphere File
#
INPUT_FILE=anc18_001_01_00.dat  0000000 0050006
#
# Geoid file
#
INPUT_FILE=anc13_001_01_00.dat  0000000 0050006
#
# Load Tide file
#
INPUT_FILE=anc16_001_01_00.dat  0000000 0050006
#
# Ocean tide file
#
INPUT_FILE=anc17_001_01_00.dat  0000000 0050006
#
#----------Dynamic ANC Files
#
# Input MET Files : 00 = hdr, 01=pwat, 02=hgt, 03=rh, 04=tmp
#
# 1st 6 hours
#
INPUT_FILE=anc01_001_20000101_120000_01_00.dat 0000000 0021599
INPUT_FILE=anc01_001_20000101_120000_01_01.dat 0000000 0021599
INPUT_FILE=anc01_001_20000101_120000_01_02.dat 0000000 0021599
INPUT_FILE=anc01_001_20000101_120000_01_03.dat 0000000 0021599
INPUT_FILE=anc01_001_20000101_120000_01_04.dat 0000000 0021599
#
# 2nd 6 hours
#
INPUT_FILE=anc01_001_20000101_180000_01_00.dat 0021600 0043199
INPUT_FILE=anc01_001_20000101_180000_01_01.dat 0021600 0043199
INPUT_FILE=anc01_001_20000101_180000_01_02.dat 0021600 0043199
INPUT_FILE=anc01_001_20000101_180000_01_03.dat 0021600 0043199
INPUT_FILE=anc01_001_20000101_180000_01_04.dat 0021600 0043199
#
# POD (precision) - Stop time MUST be > stop time of data !!!
#
INPUT_FILE=anc08_001_20000101_000000_01_00.dat 0000000 0050006
#
# PAD
#
INPUT_FILE=anc09_001_20000101_000000_01_00.dat 0000000 0050006
#
```

```
# Rotation Matrix
#
INPUT_FILE=anc24_001_20000101_120000_01_00.dat 0000000 0050006
#
#----------Input GLA Files
#
# Input GLA05 Files
#
INPUT_FILE=gla05_001_11_001_0001_1_01_00.dat 0000001 0000808
INPUT_FILE=gla05_001_11_001_0001_2_01_00.dat 0000808 0002100
INPUT_FILE=gla05_001_11_001_0001_3_01_00.dat 0002100 0003711
#
#----------Output ANC Files
#
# Output ANC06
#
OUTPUT_FILE=anc06_001_20001027_004.dat   0000000 0050006
#
#----------Output GLA Files
#
# Output GLA06
#
OUTPUT_FILE=gla06_001_11_001_0001_1_01_00.dat 0000001 0000808
OUTPUT_FILE=gla06_001_11_001_0001_2_01_00.dat 0000808 0002100
OUTPUT_FILE=gla06_001_11_001_0001_3_01_00.dat 0002100 0003711
#
# In production, GLA12-15 are not a 1-1 match with GLA06. The
# integration tests were run this way, so the acceptance
# test will be this way as well. Normally, GLA12-15 are
# not segment-dependent.
#
# Output GLA12
#
OUTPUT_FILE=gla12_001_11_001_0001_1_01_00.dat 0000001 0000808
OUTPUT_FILE=gla12_001_11_001_0001_2_01_00.dat 0000808 0002100
OUTPUT_FILE=gla12_001_11_001_0001_3_01_00.dat 0002100 0003711
#
# Output GLA13
#
OUTPUT_FILE=gla13_001_11_001_0001_1_01_00.dat 0000001 0000808
OUTPUT_FILE=gla13_001_11_001_0001_2_01_00.dat 0000808 0002100
OUTPUT_FILE=gla13_001_11_001_0001_3_01_00.dat 0002100 0003711
#
# Output GLA14
#
OUTPUT_FILE=gla14_001_11_001_0001_1_01_00.dat 0000001 0000808
OUTPUT_FILE=gla14_001_11_001_0001_2_01_00.dat 0000808 0002100
OUTPUT_FILE=gla14_001_11_001_0001_3_01_00.dat 0002100 0003711
#
# Output GLA15
#
OUTPUT_FILE=gla15_001_11_001_0001_1_01_00.dat 0000001 0000808
OUTPUT_FILE=gla15_001_11_001_0001_2_01_00.dat 0000808 0002100
OUTPUT_FILE=gla15_001_11_001_0001_3_01_00.dat 0002100 0003711
```

```
#
#----------Execution Control
#
SURFACE_TYPE=ALL
ELEVATION_PROCESS=ALL
#
#----------End of Control File
#
=New Utility
```

A.5 Test 5 Control File - Partial Elevation

```
#
#----------Start of Control File
#
# The control file for the Elevation acceptance test
#
#----------Execution Information
#
= GLAS_Exec
TEMPLATE=elev_control_file
EXEC_KEY=10023426
DATE_GENERATED=2000 October 27
OPERATOR=jlee
#
#----------Cycle Track Boundaries
#
# Cycle/Track Boundaries
#
CYCLE=001 0000000 0000060
TRACK=0001 0000001 0004998
TRACK=0002 0004998 0010798
#
#----------Static ANC Files
#
# Input ANC07 Files : 00=error, 01=global, 03=elevation
#
INPUT_FILE=anc07_001_01_00.dat  0000000 0050006
INPUT_FILE=anc07_001_01_01.dat  0000000 0050006
INPUT_FILE=anc07_001_01_03.dat  0000000 0050006
#
# Input DEM Files : 00 = DEM, 01=Mask
#
INPUT_FILE=anc12_001_01_00.dat  0000000 0050006
INPUT_FILE=anc12_001_01_01.dat  0000000 0050006
#
# Input Standard Atmosphere File
#
INPUT_FILE=anc18_001_01_00.dat  0000000 0050006
#
# Geoid file
#
INPUT_FILE=anc13_001_01_00.dat  0000000 0050006
```

```
#
# Load Tide file
#
INPUT_FILE=anc16_001_01_00.dat  0000000 0050006
#
# Ocean tide file
#
INPUT_FILE=anc17_001_01_00.dat  0000000 0050006
#
#----------Dynamic ANC Files
#
# Input MET Files : 00 = hdr, 01=pwat, 02=hgt, 03=rh, 04=tmp
#
# 1st 6 hours
#
INPUT_FILE=anc01_001_20000101_120000_01_00.dat 0000000 0021599
INPUT_FILE=anc01_001_20000101_120000_01_01.dat 0000000 0021599
INPUT_FILE=anc01_001_20000101_120000_01_02.dat 0000000 0021599
INPUT_FILE=anc01_001_20000101_120000_01_03.dat 0000000 0021599
INPUT_FILE=anc01_001_20000101_120000_01_04.dat 0000000 0021599
#
# 2nd 6 hours
#
INPUT_FILE=anc01_001_20000101_180000_01_00.dat 0021600 0043199
INPUT_FILE=anc01_001_20000101_180000_01_01.dat 0021600 0043199
INPUT_FILE=anc01_001_20000101_180000_01_02.dat 0021600 0043199
INPUT_FILE=anc01_001_20000101_180000_01_03.dat 0021600 0043199
INPUT_FILE=anc01_001_20000101_180000_01_04.dat 0021600 0043199
#
# POD (precision) - Stop time MUST be > stop time of data !!!
#
INPUT_FILE=anc08_001_20000101_000000_01_00.dat 0000000 0050006
#
# PAD
#
INPUT_FILE=anc09_001_20000101_000000_01_00.dat 0000000 0050006
#
# Rotation Matrix
#
INPUT_FILE=anc24_001_20000101_120000_01_00.dat 0000000 0050006
#
#----------Input GLA Files
#
# Input GLA05 Files
#
INPUT_FILE=gla05_001_11_001_0001_1_01_00.dat 0000001 0000808
INPUT_FILE=gla05_001_11_001_0001_2_01_00.dat 0000808 0002100
INPUT_FILE=gla05_001_11_001_0001_3_01_00.dat 0002100 0003711
#
#----------Output ANC Files
#
# Output ANC06
#
OUTPUT_FILE=anc06_001_20001027_005.dat   0000000 0050006
```

```
#
#----------Output GLA Files
#
# Output GLA06
#
OUTPUT_FILE=gla06_001_11_001_0001_1_01_00.dat 0000001 0000808
OUTPUT_FILE=gla06_001_11_001_0001_2_01_00.dat 0000808 0002100
OUTPUT_FILE=gla06_001_11_001_0001_3_01_00.dat 0002100 0003711
#
# Output GLA12
#
#
# Output GLA13
#
#
# Output GLA14
#
#
# Output GLA15
#
#
#----------Execution Control
#
ELEVATION_PROCESS=ALL
#
#----------End of Control File
#
=New Utility
```

Abbreviations & Acronyms

ALT	designation for the EOS-Altimeter spacecraft series
DAAC	Distributed Active Archive Center
EDOS	EOS Data and Operations System
EOC	EOS Operating Center
EOS	NASA Earth Observing System Mission Program
EOSDIS	Earth Observing System Data and Information System
GDS	GLAS Ground Data System
GLAS	Geoscience Laser Altimeter System instrument or investigation
GPS	Global Positioning System
GSFC	NASA Goddard Space Flight Center at Greenbelt, Maryland
GSFC/WFF	NASA Goddard Space Flight Center/Wallops Flight Facility at Wallops Island, Virginia
ID	Identification
IEEE	Institute for Electronics and Electrical Engineering
IST	GLAS Instrument Support Terminal
LASER	Light Amplification by Stimulated Emission of Radiation
LIDAR	Light Detection and Ranging
N/A	Not (/) Applicable
NASA	National Aeronautics and Space Administration
NOAA	National Oceanic and Atmospheric Administration
POD	Precision Orbit Determination
SCF	GLAS investigation Science Computing Facility and workstation(s)
SDPS	Science Data Processing Segment
TBD	to be determined, to be done, or to be developed
UNIX	the operating system jointly developed by the AT&T Bell Laboratories and the University of California-Berkeley System Division

Glossary

aggregate	A collection, assemblage, or grouping of distinct data parts together to make a whole. It is generally used to indicate the grouping of GLAS data items, arrays, elements, and EOS parameters into a data record. For example, the collection of Level 1B EOS Data Parameters gathered to form a one-second Level 1B data record. It could be used to represent groupings of various GLAS data entities such as data items aggregated as an array, data items and arrays aggregated into a GLAS Data Element, GLAS Data Elements aggregated as an EOS Data Parameter, or EOS Data Parameters aggregated into a Data Product record.
array	An ordered arrangement of homogenous data items that may either be synchronous or asynchronous. An array of data items usually implies the ability to access individual data items or members of the array by an index. An array of GLAS data items might represent the three coordinates of a georeference location, a collection of values at a rate, or a collection of values describing an altimeter waveform.
file	A collection of data stored as records and terminated by a physical or logical end-of-file (EOF) marker. The term usually applies to the collection within a storage device or storage media such as a disk file or a tape file. Loosely employed it is used to indicate a collection of GLAS data records without a standard label. For the Level 1A Data Product, the file would constitute the collection of one-second Level 1A data records generated in the SDPS working storage for a single pass.
header	A text and/or binary label or information record, record set, or block, prefacing a data record, record set, or a file. A header usually contains identifying or descriptive information, and may sometimes be embedded within a record rather than attached as a prefix.
item	Specifically, a data item. A discrete, non-decomposable unit of data, usually a single word or value in a data record, or a single value from a data array. The representation of a single GLAS data value within a data array or a GLAS Data Element.
label	The text and/or binary information records, record set, block, header, or headers prefacing a data file or linked to a data file sufficient to form a labeled data product. A standard label may imply a standard data product. A label may consist of a single header as well as multiple headers and markers depending on the defining authority.
Level 0	The level designation applied to an EOS data product that consists of raw instrument data, recorded at the original resolution, in time order, with any duplicate or redundant data packets removed.
Level 1A	The level designation applied to an EOS data product that consists of reconstructed, unprocessed Level 0 instrument data, recorded at the full resolution with time referenced data records, in time order. The data are annotated with ancillary information including radiometric and geometric calibration coefficients, and georeferencing parameter data (i.e., ephemeris data). The included, computed coefficients and parameter data have not however been applied to correct the Level 0 instrument data contents.

Level 1B	The level designation applied to an EOS data product that consists of Level 1A data that have been radiometrically corrected, processed from raw data into sensor data units, and have been geolocated according to applied georeferencing data.
Level 2	The level designation applied to an EOS data product that consists of derived geophysical data values, recorded at the same resolution, time order, and georeference location as the Level 1A or Level 1B data.
Level 3	The level designation applied to an EOS data product that consists of geophysical data values derived from Level 1 or Level 2 data, recorded at a temporally or spatially resampled resolution.
Level 4	The level designation applied to an EOS data product that consists of data from modeled output or resultant analysis of lower level data that are not directly derived by the GLAS instrument and supplemental sensors.
metadata	The textual information supplied as supplemental, descriptive information to a data product. It may consist of fixed or variable length records of ASCII data describing files, records, parameters, elements, items, formats, etc., that may serve as catalog, data base, keyword/value, header, or label data. This data may be parsable and searchable by some tool or utility program.
orbit	The passage of time and spacecraft travel signifying a complete journey around a celestial or terrestrial body. For GLAS and the EOS ALT-L spacecraft each orbit starts at the time when the spacecraft is on the equator traveling toward the North Pole, continues through the equator crossing as the spacecraft ground track moves toward the South Pole, and terminates when the spacecraft has reached the equator moving northward from the South Polar region.
model	A graphical representation of a system.
module	A collection of program statements with four basic attributes: input and output, function, mechanics and internal data.
parameter	Specifically, an EOS Data Parameter. This is a defining, controlling, or constraining data unit associated with a EOS science community approved algorithm. It is identified by an EOS Parameter Number and Parameter Name. An EOS Data Parameter within the GLAS Data Product is composed of one or more GLAS Data Elements
pass	A sub-segment of an orbit, it may consist of the ascending or descending portion of an orbit (e.g., a descending pass would consist of the ground track segment beginning with the northernmost point of travel through the following southernmost point of travel), or the segment above or below the equator; for GLAS the pass is identified as either the northern or southern hemisphere portion of the ground track on any orbit
PDL	Program Design Language (Pseudocode). A language tool used for module programming and specification. It is at a higher level than any existing compilable language.
process	An activity on a dataflow diagram that transforms input data flow(s) into output data flow(s).

product	Specifically, the Data Product or the EOS Data Product. This is implicitly the labeled data product or the data product as produced by software on the SDPS or SCF. A GLAS data product refers to the data file or record collection either prefaced with a product label or standard formatted data label or linked to a product label or standard formatted data label file. Loosely used, it may indicate a single pass file aggregation, or the entire set of product files contained in a data repository.
program	The smallest set of computer instructions that can be executed as a stand-alone unit
record	A specific organization or aggregate of data items. It represents the collection of EOS Data Parameters within a given time interval, such as a one-second data record. It is the first level decomposition of a product file.
Scenario	A single execution path for a process.
Standard Data Product	Specifically, a GLAS Standard Data Product. It represents an EOS ALT-L/GLAS Data Product produced on the EOSDIS SDPS for GLAS data product generation or within the GLAS Science Computing Facility using EOS science community approved algorithms. It is routinely produced and is intended to be archived in the EOSDIS data repository for EOS user community-wide access and retrieval.
State Transition Diagram	Graphical representation of one or more scenarios.
Stub	(alias dummy module) a primitive implementation of a subordinate module, which is normally used in the top-down testing of superordinate (higher) modules.
Structured Chart	A graphical tool for depicting the partitioning of a system into modules, the hierarchy and organization of those modules, and the communication interfaces between the modules.
Structured Design	The development of a blueprint of a computer system solution to a problem, having the same components and interrelationships amount the components as the original problem has.
Subroutine	A program that is called by another program
variable	Usually a reference in a computer program to a storage location, i.e., a place to contain or hold the value of a data item.